REMARKS

Claims 1 to 8 were pending in the application at the time of examination. The Examiner rejected Claims 1 to 8 under 35 U.S.C.103(a) as obvious over the Chopra et al. reference (US 6,167,423) in view of the Cutler et al. reference (US 5,057,996).

Applicant has cancelled Claims 2 and 3, without prejudice. Applicant has amended Claims 1, 4, 7 and 8. Consequently, Claims 1 and 4 to 8 remain in the application.

REJECTION OF CALIMS 1 TO 8 UNDER 35 U.S.C.103(a)

The Examiner rejected Claims 1 to 8 under 35 U.S.C.103(a) as obvious over the Chopra et al. reference (US 6,167,423) in view of the Cutler et al. reference (US 5,057,996).

Applicant has cancelled Claims 2 and 3, without prejudice, and incorporated the limitations of Claims 2 and 3 in Claim 1, as amended. Consequently, Applicant respectfully submits that the rejection of Claims 2 and 3 is now moot.

As shown above, Applicant has amended independent Claim 1. Claim 1, as amended, recites, with emphasis added:

> A data structure comprising:
> a **combined thread control block/message structure**, said combined thread control block/message structure comprising:
> a thread control block structure, **said thread control block structure including a message structure embodied therein**, said thread control block structure configured to store information used to control execution of a thread, said message structure configured to store a message.

GUNNISON, McKAY & HODGSON, L.L.P.
Garden West Office Plaza
1900 Garden Road, Suite 220
Monterey, CA 93940
(831) 655-0880
Fax (831) 655-0888

The advantages of the combined thread control block/message structure of Applicant's Claims is discussed in Applicant's Specification at pages 3 to 4, pages 8 and 9 and pages 21 to 27. For instance page 8 line 13 to page 9, line 17 of Applicant's Specification reads as follows with emphasis added:

> **By combining a TCB's data structure and a message's data structure, an operating system employing an embodiment of the present invention can be constructed more simply and so operate more efficiently. An operating system incorporating an embodiment of the present invention is simplified because of simplified error handling, reduced indirection, fewer initial allocations and reduced allocation/de-allocation operations, among other advantages, as explained previously.**
>
> Such an operating system's efficiency is also improved by the ability of a combined TCB/message structure to control both thread execution and message passing. For example, upon the receipt of a TCB/message structure, a task has all the control information necessary to perform information transfer and thread control. In one configuration, for example, queuing a TCB/message structure to an I/O channel (as described subsequently herein) of a task not only provides the task's thread with the requisite information, but also provides thread control information to the task, obviating the need for the task or operating system to coordinate message information with task information. The TCB/message structure provides the task with the control information needed to access the information associated with the message. The TCB/message structure also provides the task with thread control information, allowing the task to start (or re-start) execution of the given thread at the appropriate time. For example, the thread control information held in the TCB/message structure can be used to cause execution of the thread to begin only after the

GUNNISON, McKAY & HODGSON, L.L.P.
Garden West Office Plaza
1900 Garden Road, Suite 220
Monterey, CA 93940
(831) 655-0880
Fax (831) 655-0888

message passing operation has completed. Thus, the task need only access one structure to acquire all the information necessary to both complete the message passing operation and control the thread associated therewith.

Moreover, because such combined structures are pre-allocated in an operating system according to the present invention, failure due to allocation errors is avoided. In other words, if there will not be enough memory to create a thread control block, that fact will become apparent when the pre-allocation is performed. As noted, this is of particular importance in a mission-critical system because the occurrence of such a dynamic failure during a dynamic allocation would likely cause an operating system to fail, and because such failures are non-deterministic in nature (and so cannot be predicted with any accuracy), they are especially dangerous in mission-critical systems. This aspect also avoids the operating system becoming deadlocked during I/O operations, waiting for the allocation of a message that cannot be allocated due to a lack of memory space.

In addition, page 22, line 4 to page 27 line 2 of Applicant's Specification reads as follows with emphasis added:

Fig. 8B is a block diagram illustrating data structures of the prior art. An I/O packet (e.g., I/O packet 840), which provides for the transfer of data within an operating system, has historically been intended and so designed as a data structure separate from the data structure used to control a thread (e.g., thread control block 850). Originally, there was no need for a thread control block because the concept of a multi-threaded task did not exist, and so only I/O operations required a structure definition. **More recently, with the advent of threads, the thread control block structure and the message structure were viewed as separate entities because the operations performed by each were conceptualized as being distinct and separate. Thus,**

GUNNISON, McKAY & HODGSON, L.L.P.
Garden West Office Plaza
1900 Garden Road, Suite 220
Monterey, CA 93940
(831) 655-0880
Fax (831) 655-0888

a thread control block such as a thread control block 840 was designed as a separate structure to address the provision of I/O facilities to the various elements of the given operating system and user applications run thereon. In a similar fashion, a thread control block such as thread control block 850 was specifically designed to control the execution of threads (e.g., threads within user applications), without any thought to the mechanisms used for I/O communications. The two structures might reference one another, but their diverging applications did not obviously lend themselves to an integrated structure. For example, a reference 852 allows thread control block 850 to monitor I/O packet 840, as a reference 854 allows I/O packet 840 to monitor thread control block 850. However, this falls far short of integrating the two structures, and makes apparent the fact that neither of thread control block 850 or I/O packet 840 is designed to work cooperatively with the other. This conceptual separation provided greater simplicity for the two structures themselves, but failed to address the needs of a microkernel operating system implementation, including simplicity and efficiency of the microkernel, simplicity of the message passing/thread execution paradigm, and the like.

Fig. 8C is a block diagram illustrating a combined TCB/message data structure 860. As can be seen, the data structure of a message 870 is now integrated into the data structure of a thread control block 880. Programmatically, this might be structured as follows:

```
    /*-----------------------------------------------
--------------
                      The MSG and DDR structures
    -----------------------------------------------
------------*/

    #define DDRINLINEBUFSIZE  96
    #define DDR2INLINEBUFSIZE 64
    #define DDR1INLINEBUFSIZE 32
    #define DDR0INLINEBUFSIZE 0
```

GUNNISON, McKAY & HODGSON, L.L.P.
Garden West Office Plaza
1900 Garden Road, Suite 220
Monterey, CA 93940
(831) 655-0880
Fax (831) 655-0888

```
        typedef struct mrb_t {
                short           r_fidOrMid;
                uchar           r_op;
                uchar           r_ctl;
                int             r_result;
                union {
                        int             x_arg;          /*
1 32 bit args */
                        int             x_args[2];      /*
2 32 bit args */
                        short           x_shortargs[4]; /*
4 16 bit args */
                        longlong        x_bigarg;       /*
1 64 bit arg  */
                } u;
        } MRB;

        #define r_arg       u.x_arg
        #define r_args      u.x_args
        #define r_bigarg    u.x_bigarg
        #define r_shortargs u.x_shortargs

        #define MSGMAIN     \

                MRB         m_mrb;     /* Input and
Output parameters */

        #define m_fidOrMid  m_mrb.r_fidOrMid
        #define m_op        m_mrb.r_op
        #define m_ctl       m_mrb.r_ctl


        /*----------------------------------------*/
        /*      Macros                            */
        /*----------------------------------------*/

        #define m_fid       m_fidOrMid
        #define m_mid       m_fidOrMid
        #define m_result    m_mrb.r_result
        #define m_cid       m_mrb.r_result
        #define m_args      m_mrb.r_args
        #define m_arg       m_mrb.r_arg
        #define m_bigarg    m_mrb.r_bigarg
        #define m_shortargs m_mrb.r_shortargs
        #define m_type      m_ddr.d_type
        #define m_inline    m_ddr.d_inline
        #define m_pid       m_ddr.d_pid
        #define m_base      m_ddr.d_base
        #define m_offset    m_ddr.d_offset
```

GUNNISON, McKAY &
HODGSON, L.L.P.
Garden West Office Plaza
1900 Garden Road, Suite 220
Monterey, CA 93940
(831) 655-0880
Fax (831) 655-0888

```
#define m_length    m_ddr.d_length
#define m_inlinebuf m_ddr.d_inlinebuf


#define DDRMAIN    \
   char   d_type;      /* Type of DDR DT_XXXX
      */ \
   char   d_inline;    /* 0, 1, 2, or 3 groups of
32 bytes  */ \
   short  d_ctx;       /* Context of DDR **
SYSTEM FIELD ** */ \
   void  *d_base;      /* Ignored if d_inline !=
0          */ \
   int    d_offset;    /* Unused for inline or
DT_VADDRESS  */ \
   int    d_length;


/*-------------------------------------------*/
/* DDR and MSG with 96 bytes of inline data */
/*-------------------------------------------*/

typedef struct ddr_t3 {
        DDRMAIN
        char   d_inlinebuf[DDRINLINEBUFSIZE];
} DDR, DDR3;

typedef struct msg_t3 {
        MSGMAIN
        DDR      m_ddr;
} MSG3, MSG;


/*-------------------------------------------*/
/* DDR and MSG with 64 bytes of inline data */
/*-------------------------------------------*/

typedef struct ddr_t2 {
        DDRMAIN
        char   d_inlinebuf[DDR2INLINEBUFSIZE];
} DDR2;

typedef struct msg_t2 {
        MSGMAIN
        DDR2    m_ddr;
} MSG2;
```

GUNNISON, McKAY &
HODGSON, L.L.P.
Garden West Office Plaza
1900 Garden Road, Suite 220
Monterey, CA 93940
(831) 655-0880
Fax (831) 655-0888

```
/*-------------------------------------------*/
/* DDR and MSG with 32 bytes of inline data */
/*-------------------------------------------*/

typedef struct ddr_t1 {
        DDRMAIN
        char   d_inlinebuf[DDR1INLINEBUFSIZE];
} DDR1;

typedef struct msg_t1 {
        MSGMAIN
        DDR1    m_ddr;
} MSG1;


/*------------------------------------------*/
/* DDR and MSG with 0 bytes of inline data */
/*------------------------------------------*/

typedef struct ddr_t0 {
        DDRMAIN
        char   d_inlinebuf[DDR2INLINEBUFSIZE];
} DDR0;

typedef struct msg_t0 {
        MSGMAIN
        DDR0    m_ddr;
} MSG0;


/*----------------------------------------------
--------------
                           TCB Structure
----------------------------------------------
------------*/

typedef struct tcb_t {

        LEB               t_list;
        int               t_index;
        int               t_type;
        int               t_inputP3;

        struct cpu_t     *t_copyTaskCpu;
        struct cpu_t     *t_cpu;
        struct pcb_t     *t_pcb;
        struct pcb_t     *t_queuedTo;

        int               t_state;
```

GUNNISON, McKAY &
HODGSON, L.L.P.
Garden West Office Plaza
1900 Garden Road, Suite 220
Monterey, CA 93940
(831) 655-0880
Fax (831) 655-0888

```
                int             t_dirtyRegs;
                int             t_priority;
                int             t_lastPrioritySetting;


                int             t_semaphore;
                boolean         t_hasLockedRegs;

                CLK             t_clockBlock;
                boolean         t_swappingWasEnabled;

                int             t_handler;

                MSG             t_message;

                char
t_stackEnd[KERNELSTACKSIZE-sizeof(TRP)];

                TRP             t_trapframe;

                int             t_kstackStart;
        } TCB;
```

As can be seen in the structure definitions above, a message structure (of type MSG) is included as part of each thread control block structure (of type TCB). As can also be seen in the structure definitions above, a combined TCB/message structure according to an embodiment of the present invention includes both thread information and message information. Access to both thread information and message information is simplified because a level of indirection is avoided through the use of a combined TCB/message structure. The thread information relates mostly to the control of the particular thread in question, and includes information such as information regarding the process and CPU on which the thread is running, the thread's state, the thread's environment, register information, FPU information, and stack frame and trap frame information. The message information relates mostly to the transfer of data to or from the particular thread in question, and includes information such as type information (indicating the data structured used to transfer data to the receiving task), in-line data information (as to whether the data being transferred (if any) is in-line, and optionally the size of the data field

GUNNISON, McKAY &
HODGSON, L.L.P.
Garden West Office Plaza
1900 Garden Road, Suite 220
Monterey, CA 93940
(831) 655-0880
Fax (831) 655-0888

> **used), context information, base address information, offset information and length information. Other information and structures can be supported by a combined TCB/message structure according to embodiments of the present invention, as evidenced by the above program listing. As will be apparent to one of skill in the art, the actual structure definition of the message could be included in the thread control block structure (i.e., the code listing could be structured such that the thread control block's definition includes the message structure's definition), but this would make reading the listing of the thread control block's structure definition unnecessarily complicated. By integrating the message structure into the thread control block structure, an operating system benefits from the advantages of a combined TCB/message structure according to an embodiment of the present invention previously enumerated.**

In making the rejection of Claim 1, the Examiner equates the clique structure of Chopra with the combined thread control block/message structure of Applicant's invention and the message queue of Chopra with the message structure of Applicant's claims. The Examiner then notes that the Chopra structure still lacks the thread control block of Applicant's invention and so combines the structure of Chopra with Cutler. The Examiner then states that under very specific instances the proposed combination would behave similarly to Applicant's invention.

Applicant respectfully traverses this argument. Applicant first notes that the structure of Chopra uses Cliques, which are:

> ...a collection of connections that deliver a single message at a time in a serialized fashion.
>
> (Chopra, column 3, lines 8 to 10)

GUNNISON, McKAY &
HODGSON, L.L.P.
Garden West Office Plaza
1900 Garden Road, Suite 220
Monterey, CA 93940
(831) 655-0880
Fax (831) 655-0888

According to Chopra:

> In accordance with the invention, connections to a state machine or set of state machines with private data in shared memory or other need for concurrency isolation are grouped into collections called "cliques". A clique is defined as a collection of connections that deliver a single message at a time in a serialized fashion. A system or method (e.g., a connection manager in the illustrated embodiment that manages message delivery to the cliques ensures serialized delivery of messages to each clique. As further discussed below, the connection manager also ensures that only a single thread at a time execute in the state machines within a clique.

(Chopra, column 3, lines 5 to 15)

Applicant respectfully submits that the structure disclosed in Chopra is similar to the conceptually separated structures discussed in Applicant's specification at page 22 lines 10 to 28, which reads as follows:

> More recently, with the advent of threads, the thread control block structure and the message structure were viewed as separate entities because the operations performed by each were conceptualized as being distinct and separate. Thus, a thread control block such as a thread control block 840 was designed as a separate structure to address the provision of I/O facilities to the various elements of the given operating system and user applications run thereon. In a similar fashion, a thread control block such as thread control block 850 was specifically designed to control the execution of threads (e.g., threads within user applications), without any thought to the mechanisms used for I/O communications. The two

GUNNISON, McKAY &
HODGSON, L.L.P.
Garden West Office Plaza
1900 Garden Road, Suite 220
Monterey, CA 93940
(831) 655-0880
Fax (831) 655-0888

> structures might reference one another, but their diverging applications did not obviously lend themselves to an integrated structure. For example, a reference 852 allows thread control block 850 to monitor I/O packet 840, as a reference 854 allows I/O packet 840 to monitor thread control block 850. However, this falls far short of integrating the two structures, and makes apparent the fact that neither of thread control block 850 or I/O packet 840 is designed to work cooperatively with the other. This conceptual separation provided greater simplicity for the two structures themselves, but failed to address the needs of a microkernel operating system implementation, including simplicity and efficiency of the microkernel, simplicity of the message passing/thread execution paradigm, and the like.

Applicant respectfully submits that the Examiner has failed to show where in the Chopra et al. reference, the Cutler et al. reference, or any proper combination of the Chopra et al. reference and the Cutler et al. reference, there is disclosed the combined thread control block/message structure specifically recited in Applicant's Claim 1, as amended.

In light of the discussion above, Applicant respectfully submits that Claim 1, as amended, is patentable over the Chopra et al. reference, the Cutler et al. reference, or any proper combination of the Chopra et al. reference and the Cutler et al. reference, for at least the reasons discussed above. Consequently, Applicant respectfully requests the Examiner withdraw the rejection of Claim 1 under 35 U.S.C.103(a) and allow Claim 1 to issue.

Claims 4 to 8 depend, directly or indirectly on Claim 1, as amended. Consequently, Claims 4 to 8 include all of the features and limitations of Claim 1, as amended. Therefore Applicant respectfully submits that Claims 4 to 8 are patentable over the Chopra et al. reference, the Cutler et al. reference, or any proper combination of the Chopra et al. reference and the Cutler et al. reference, for at least the

GUNNISON, McKAY &
HODGSON, L.L.P.
Garden West Office Plaza
1900 Garden Road, Suite 220
Monterey, CA 93940
(831) 655-0880
Fax (831) 655-0888

reasons discussed above. Consequently, Applicant respectfully requests the Examiner withdraw the rejection of Claims 4 to 8 under 35 U.S.C.103(a) and allow Claims 4 to 8 to issue as well.

CONCLUSION

For the foregoing reasons, Applicant respectfully requests allowance of all pending Claims 1 and 4 to 8, as amended. If the Examiner has any questions relating to the above, the Examiner is respectfully requested to telephone the undersigned Attorney for Applicant.

CERTIFICATE OF MAILING
I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on **July 12, 2004**.

[illegible]
Attorney for Applicant(s)

July 12, 2004
Date of Signature

Respectfully submitted,

[illegible]

Philip J. McKay
Attorney for Applicant
Reg. No. 38,966
Tel.: (831) 655-0880

GUNNISON, McKAY & HODGSON, L.L.P.
Garden West Office Plaza
1900 Garden Road, Suite 220
Monterey, CA 93940
(831) 655-0880
Fax (831) 655-0888